

Physically Realistic Morphing

Jessica K. Hodgins and Nancy S. Pollard

College of Computing and Graphics, Visualization and Usability Center

Georgia Institute of Technology

Atlanta, GA 30332-0280

[jkh|nsp]@cc.gatech.edu

Abstract

This paper describes an algorithm for automatically adapting existing animated behaviors to new actors with different physical characteristics. When simulation is used for animation, this is a difficult problem because a control system that is tuned for one actor will not work on an actor with different limb lengths, masses, or moments of inertia. The algorithm presented here adapts the control system to a new actor in two stages. First, the control system parameters are scaled based on the size and moment of inertia of the dynamic models for the new and the old actors. Then a subset of the parameters is fine-tuned using a search process based on simulated annealing. To demonstrate the effectiveness of this approach, we animate the running motion of a woman and child by modifying the control system for a running man. In addition to adapting a control system for a new model, this approach can also be used to adapt the control system in an on-line fashion to produce a physically realistic metamorphosis from the old model to the new model while the morphing model is performing the behavior. We evaluate the results of this approach by comparing the motion of the simulated child and woman with that of actual humans and data published in the literature.

Key Words: Animation Techniques (control, physics-based, kinematics, dynamics), Simulation

Additional Key Words and Phrases: human motion, motion control, dynamic simulation, simulated annealing

Introduction

If simulated humanlike characters are to be useful in animations and virtual environments, we must be able to create new, appealing actors easily. Appealing human motion has several components: the kinematics and dynamics of the figure must be physically correct

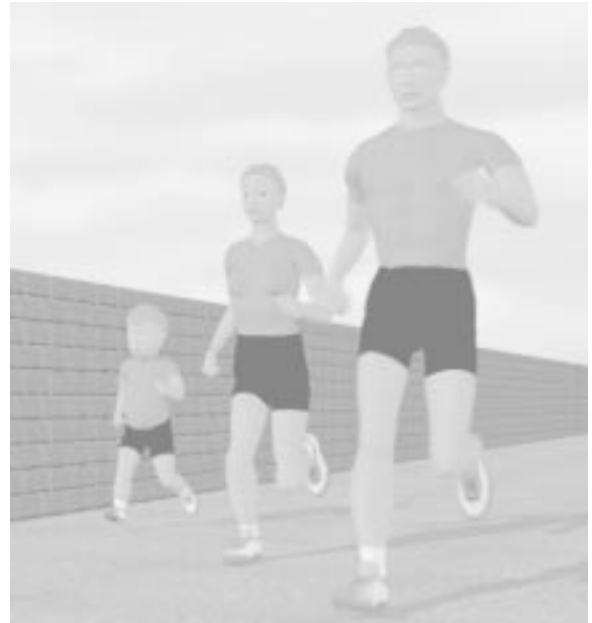


Figure 1: Image of running child, woman, and man.

and the control algorithms must make the figure perform in ways that appear natural and are stylistically appropriate for the setting and character. In this paper, we describe an algorithm for adapting existing control systems to new dynamic models to facilitate the rapid creation of new actors. We demonstrate that the control algorithm that produces running motion for a male figure can be adapted to control the running motion for a woman or child. Figure 1 shows the running motion of the three different figures. The three figures have markedly different dynamic properties and a simple, or geometric scaling is not adequate to transform one model or control system into another. For example, the man and the child differ not only in height but also in their proportions because the child has a proportionally heavier torso and shorter arms and legs.

An algorithm such as the one described in this paper should allow an animator to develop a new character by first using a commercial modeling package to de-



Figure 2: Images of the metamorphosis of a man into a woman while running. The metamorphosis occurred over a 20 second period and the images are spaced by 3 seconds.

fine the shape of the body parts and then automatically adapting an existing behavior to animate the new model. The animation process proceeds in several stages. First, a polygonal representation of the new model is used to compute the volume and mass distribution of each body part. An approximation to the new control system is obtained by scaling based on the size and moment of inertia of the new and the old models. A search process is used to fine tune the new control system. When the old and new models differ substantially, the transformation may require adapting the control system first to one or more intermediate models rather than moving directly from the old model to the new model.

This algorithm can be used not only to adapt a control system to a new model but also to perform a physically realistic metamorphosis between two models. In the transformation, the graphical model, the dynamic system, and the control system are all interpolated online while the character performs the behavior. The trajectories of the control system parameters for the transformation are found using a two stage process of scaling and tuning that is similar to the procedure used for the off-line adaptation. Figure 2 shows the metamorphosis of a man into a woman while running.

Much of the style of the three running motions is driven by the physical model. For example, the running child has a proportionally wider stance than the woman because the ratio of the hip spacing to leg length is larger for the child. However, some aspects of the style are determined by the parameters of the control system, and the evaluation function of the search procedure provides a way to influence the running style. For example, the duration of the flight phase can be increased and the speed decreased to provide a casual jogger instead of a more determined runner.

Background

Generating appealing motion is the central problem in animation. Dynamic simulation and other procedural approaches are one potential solution to this problem. Simulation guarantees physical realism (or adherence to any other set of consistent “pseudo-physical” laws) but the design of control systems for characters with interesting complexity has proved difficult. Control systems or procedural algorithms can be hand-designed[12, 23, 6, 8, 13] but this approach is labor intensive and requires that the animator or programmer have extensive knowledge about the details of the behavior.

A more appealing approach is automatic or semi-automatic design. One automatic approach treats the problem of generating motion as a trajectory optimization problem. Witkin and Kass[36] used this approach to control a jumping Luxo lamp. Cohen and Liu[9, 18] modified the optimization problem to have several windows thereby providing more control for the animator and reducing the time required for the optimization. They used this approach to control a two link acrobot and a planar diving figure. Liu, Gortler, and Cohen [19] implemented a hierarchical wavelet-based version of spacetime constraints to allow finer detail where necessary without increasing the computation cost uniformly. Zhao and his colleagues at the University of Pennsylvania [37] represented the trajectory of the control variables as a spline and then optimized the locations of the control points for the spline. They used this approach to control a simple planar human figure performing a vertical jump.

A second approach to automatically generating motion uses techniques from optimal control to find a control algorithm instead of a desired trajectory. Once found, control algorithms have the advantage that different but similar motions can be generated without further optimization to respond to a disturbance or interaction. However, as optimization techniques for complex systems approach real-time, this distinction

may become less significant because trajectories can be computed for new situations as they arise. In the most general case, an optimal controller must contain information on how to get from every state of the system to every other state. This problem is of higher dimension than the problem of finding a trajectory that reaches a particular goal state from a particular start state. As a result of the high dimensionality of the problem, optimal control approaches have focused on simple systems, problem domains where the space is dense with solutions, or techniques that allow the space to be represented without a fine discretization.

The first paper to introduce optimal control to the graphics community was Brotman and Netravali[5]. They used a linear quadratic regulator to control the motion of a single body on the plane. Huang and van de Panne[14] used a best-first search to discover a sequence of set points that, when combined with a proportional-derivative servo, allowed a two link acrobot to hop and flip. van de Panne and his colleagues[34, 31] synthesized closed loop controllers for the jumping Luxo lamp and other simple systems using dynamic programming. Fiume and van de Panne[33] used a generate and test strategy to produce neural network-based control systems for a wide variety of planar creatures with 3-6 links. Ngo and Marks and their colleagues[21, 1] used a similar generate and test approach to find stimulus/response systems that animate a variety of behaviors for planar and three-dimensional figures. Sims[28, 27] used genetic algorithms to find both creatures constructed of linked bodies and competitive behaviors for the task of capturing a block.

Automatic techniques that begin the optimization or search process without significant knowledge of the behavior have not yet been successfully used for complex models such as three-dimensional humanlike figures. For a human figure with a realistic number of degrees of freedom, the search space is substantial and the density of acceptable solutions is low. However, knowledge about the behavior can be used to focus the search. Knowledge can be incorporated in the form of external guiding forces, via an existing but imperfect control system, or through motion capture data. Lamouret and van de Panne[32] used external forces to maintain the attitude of the body of a walking human figure, thereby guiding the optimization process towards the desired solution. The external force was eliminated by later stages of the optimization. Laszlo[16] used limit cycle control to stabilize open-loop trajectories for human walking of a model with 19 degrees of freedom. Ringrose adapted the control system of a planar quadruped to carry additional weight, have longer leg lengths or heavier feet[24]. In this paper, we take a similar approach by automatically adapting an existing control system for a new dynamic

model. Because our control system is more complex and has more parameters, we include more extensive knowledge about the behavior in form of a priori scaling and parameter selection. We believe that incorporating this knowledge improves the resulting control system and allows us to use fewer intermediate models for transitioning between more complex systems.

Other researchers have realized that if generating motion directly is too difficult, perhaps we can adapt existing motion trajectories from procedural approaches, simulation, or motion capture to new situations[35, 7, 30]. Rose and his colleagues at Microsoft[25] use inverse kinematics, dynamic constraints, and optimization to blend between motion capture sequences for a three-dimensional human model. Techniques for adapting existing trajectories via optimization share with the work described here the idea that optimization procedures can be used to adapt to new situations. The two approaches differ in the level of the parameters used by the optimization. Motion trajectories contain little explicit knowledge of the task being performed, while a well-parameterized control system contains extensive knowledge about the task.

We also draw from research in biomechanics for data and inspiration in this work. McMahon's elegant work on scaling between species motivated the idea that the control parameters could be approximately scaled based on knowledge about the dynamics of the system[20]. The biomechanics literature also contains data about the running motion of men, women, and children and we use this data as a point of comparison for the simulated running motion in the last section of the paper. Finally, the biomechanics literature provides data on the anthropomorphic parameters of men, women, and children of various ages that we used in developing the models. In the computer graphics area, this data has been used extensively in the Jack system developed at the University of Pennsylvania. This system contains kinematic and dynamic models of humans based on biomechanical data[2] and allows the construction of models of various anthropomorphic dimensions. Jack has been used extensively for ergonomic analysis and human factors engineering as well as distributed interactive simulation.

In the remainder of the paper, we describe the algorithm for adapting control systems for steady-state running and physically realistic morphing. First, we briefly describe the dynamic models and the control system for running. The algorithm for adapting the control system through scaling and tuning is described next, followed by an analysis of the performance of the algorithm.

Measurement	Man	Woman	Child
Mass (kg)			
head	5.88	4.22 (0.72)	5.63 (0.96)
torso	26.41	17.62 (0.67)	8.47 (0.32)
pelvis	13.41	11.08 (0.83)	3.71 (0.28)
upper leg	5.94	5.08 (0.86)	1.95 (0.33)
lower leg	3.52	2.49 (0.71)	0.78 (0.22)
foot	1.12	0.67 (0.60)	0.22 (0.20)
toe	0.17	0.08 (0.47)	0.03 (0.18)
upper arm	2.22	1.41 (0.64)	0.64 (0.29)
lower arm	1.10	0.79 (0.72)	0.43 (0.39)
hand	0.55	0.31 (0.56)	0.13 (0.24)
Length (m)			
height	1.785	1.627 (0.91)	1.078 (0.60)
leg length	0.788	0.719 (0.91)	0.435 (0.55)
arm length	0.537	0.525 (0.98)	0.326 (0.61)
hip spacing	0.198	0.152 (0.77)	0.133 (0.67)
MOI ($kg\ m^2$)			
leg/hip	2.061	1.168 (0.57)	0.132 (0.064)
foot/ankle	0.014	0.006 (0.45)	0.001 (0.057)
body/ankle	76.800	45.400 (0.59)	10.100 (0.130)
body/hip	4.432	4.432 (0.57)	1.186 (0.270)

Figure 3: Measurements of the mass and size of body parts for the man, woman, and child. The numbers in parentheses show the ratio of each parameter for the woman and child to the same parameter for the man. Moment of inertia parameters (MOI) express moment of inertia of a section of the body about a particular joint.

Dynamic Simulation

The running motion described in this paper is computed using dynamic simulation. Each simulation contains the equations of motion for a rigid-body model of a human, constraint equations for the interaction between the feet and the ground, parameterized control algorithms for running, a graphical image for viewing the motion, and a user interface for changing the parameters of the simulation. During each simulation time step, the control algorithm computes desired positions and velocities for each joint based on the state of the system and the requirements of the task as specified by the user. Proportional-derivative servos compute joint torques based on the desired and actual value of each joint. The equations of motion of the system are integrated forward in time taking into account the internal joint torques and the external forces and torques from interactions with the ground plane or other objects. The details of the model and an overview of the control algorithm are given below. For the details of the control system see [13].

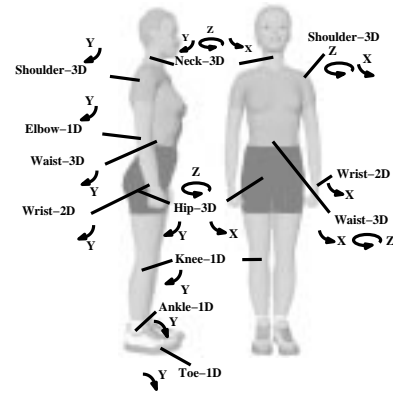


Figure 4: The controlled degrees of freedom of the three models. Each model has 17 body segments and 30 controlled degrees of freedom. The direction of the arrows indicates the positive direction of rotation for each degree of freedom. The polygonal models were created by modifying models purchased from Viewpoint Datalabs.

Dynamic Models

The human models we used to animate the running motion were constructed from rigid links connected by rotary joints with one, two or three degrees of freedom. The dynamic models were derived from the graphical models shown in figure 4 by computing the mass and moment of inertia of each body part using algorithms for computing the moment of inertia of a polygonal object of uniform density[17] and density data measured from cadavers[10]. The mass parameters of the three models are given in figure 3.

The controlled degrees of freedom of the models are shown in figure 4. Each internal joint of the model has a torque source that allows the control algorithms to apply a torque between the two links that form the joint. The equations of motion for each system were generated using a commercially available package[26, 29]. The points of contact between the feet and the ground are modeled using constraints stabilized with Baumgarte stabilization[3].

Running Control Algorithms

Running is a cyclic behavior in which the legs swing fore and aft and provide support for the body in alternation. Because the legs perform different functions during the phases of the locomotion cycle, the muscles are used for different control actions at various times in the cycle. When the foot of the simulated runner is on the ground, the ankle, knee, and hip provide support and balance. During the flight phase, a leg is swung forward in preparation for the next touchdown. These

distinct phases and corresponding changes in control actions make a state machine a natural tool for selecting the control actions that should be active at a particular time. The states correspond to the points of contact with the ground: flight, heel contact, heel and metatarsus contact, and metatarsus contact. Transitions between states are based on events such as contact of the heel or metatarsus with the ground or bending of the knee joint. To generate steady-state running, the control system must maintain three parameters: forward speed, flight duration, and balance. Each state includes control laws that compute desired values for each joint with the goal of controlling those three parameters of the running cycle.

During flight, one leg is swung forward in anticipation of touchdown. The foot is positioned at touchdown to correct for errors in forward speed and to maintain balance. The disturbances caused by the impact of touchdown can be reduced by decreasing the relative speed between the foot and the ground at touchdown. This technique is commonly called *ground speed matching*. In this control system, ground speed matching is accomplished by swinging the hip further forward in the direction of travel during flight and moving it back just before touchdown.

Flight duration is controlled by extending the ankle and knee joints during stance, causing the heel to lift off the ground and adding energy to the system for the next flight phase. Thrust is initiated when the metatarsus is in contact with the ground and the hip has moved a certain distance in front of the foot.

Throughout stance, proportional-derivative servos are used to compute torques for the hip joint of the stance leg that will cause the attitude of the body (roll, pitch, and yaw) to move toward the desired values. The desired angle for pitch is inclined slightly forward and the desired angles for roll and yaw are zero.

The control laws for forward velocity, flight duration, and balance result in desired values for each joint. Proportional-derivative servos are used to control the position of all joints. For each internal joint the control equation is

$$\tau = k(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \quad (1)$$

where θ is the angle of the joint, θ_d is the desired angle, $\dot{\theta}$ is the velocity of the joint, $\dot{\theta}_d$ is the desired velocity, and k and k_v are the proportional and derivative gains.

Scaling and Tuning

For motions as complex as human running, the parameters of a control system must be carefully tuned to match the physical characteristics of each character. A

control system that has been tuned for one dynamic model will not, in general, work on a different dynamic model. Figure 5, for example, shows the result of using the controller designed for the man in figure 1 to control the running motion for a model that is halfway between the man and the woman. The model fails to run because the control system gains and the desired velocity that are appropriate for the man model are too high for the smaller, lighter intermediate model. This section describes how the control system of one character can be adapted to produce similar motion in a different character through a two step process involving scaling and search. The scaling of the control system is accomplished based on the size, mass, and moment of inertia of the new model.

Geometric Scaling

Control systems for geometrically similar characters can be scaled based on size alone[23]. Two characters are geometrically similar if the model for one character can be obtained by scaling the length and width of the other model by a constant factor. For example, a geometrically scaled woman runner that was the same height as the child could be obtained by scaling the model of the woman by the ratio of the heights of the two models. A control system for the scaled woman could then be obtained by applying the geometric scaling rules in figure 6, using 0.66 for the scaling factor L .

The scaling rules in figure 6 allow us to adapt the parameters of the control system to account for the physical differences between two geometrically similar characters. For running, we apply geometric scaling to the following parameters:

- The state of the system (position and orientation of the torso, joint angles and the derivatives of those quantities)
- Gains for all proportional-derivative joint servos (torsional stiffness k and damping k_v in equation 1)
- Desired values used to control the running motion such as desired forward velocity, flight duration, and pitch angle
- Constants referenced by the control system such as the desired clearance of the foot during flight
- The integration time step for the dynamic simulation

Applying the scaling rules in figure 6 results in a running motion for the geometrically scaled character that is dynamically similar to that of the original.

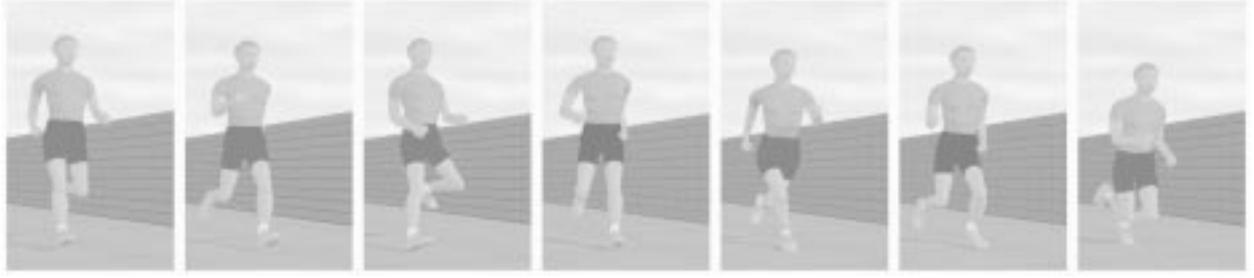


Figure 5: Images showing the result of using the control system designed for the man to control the running motion for a model that is halfway between the man and the woman. The images show successive touchdowns and are samples taken at 1.8, 2.17, 2.53, 2.8, 3, 3.53 seconds.

The running strides of the two runners are identical in appearance: any single frame taken from the running motion of one runner can be duplicated by taking a frame from the other runner and reducing or enlarging that frame. The stride frequencies will be different, however, since the number of strides per second scales as $L^{-1/2}$.

In general, two characters will not be geometrically similar. Figure 7 demonstrates this by showing a woman geometrically scaled to the height of a 3-year-old child. Figure 3 further illustrates this point by giving the mass parameters for a man, woman and child. However, geometric scaling can serve as a good first approximation, capturing some of the physical differences between two characters. For example, the model in Figure 5 runs for less than 4 seconds when the control system has not been scaled, but geometric scaling is sufficient to give it a stable running gait (i.e. the model runs for at least 100 seconds without falling).

Because geometric scaling is approximate for the models that we would like to animate, we have to choose an appropriate scaling factor. One possible choice is the overall height of the body. This choice results in scaling factors of 0.91 for the man to the woman and 0.66 for the woman to the child. A second choice is the ratio of the leg lengths, resulting in scaling factors of 0.91 for the man to the woman and 0.61 for the woman to the child. We chose to use leg length because the control and appearance of the running motion depend much more on the lower body than on the upper body and because this ratio worked better in practice.

Mass Scaling

An additional scaling step is needed to adapt the control system to physical differences not captured through geometric scaling. Figure 7 and 3, for example, show that the child is very different from a geometrically scaled version of the woman. In particular, the child

has a proportionately heavier torso and head, and relatively short arms and legs. Differences such as these are sufficient to prevent a geometrically scaled control system for the woman from producing steady-state running in the child. For example, if the leg is unexpectedly heavy, the control system will not produce sufficient torque at the hip to swing the leg forward by touchdown. The appropriate hip gains during flight depend both on the length of the leg and on the moment of inertia of the upper and lower leg and foot.

Scaling rules can be used to account for relative differences in moments of inertia. For a system with one rigid body and one angular joint, torque τ_0 at the joint produces angular acceleration $\ddot{\theta}$ as follows:

$$\tau_0 = I_0 \ddot{\theta} \quad (2)$$

where I_0 is the moment of inertia of the body about the axis of rotation. This joint is controlled with gains k_0 and k_{v0} as in equation 1:

$$\tau_0 = k_0(\theta_d - \theta) + k_{v0}(\dot{\theta}_d - \dot{\theta}) \quad (3)$$

Based on equations 2 and 3, a second system with moment of inertia I_1 and the same link lengths as the first system could be controlled to have the same angular positions, velocities, and accelerations over time by adjusting the gains as follows:

$$k_1 = k_0 \left(\frac{I_1}{I_0} \right) \quad (4)$$

$$k_{v1} = k_{v0} \left(\frac{I_1}{I_0} \right) \quad (5)$$

Given appropriate ratios for the moment of inertia terms, equations 4 and 5 express the scaling relationships required to account for differences in moments of inertia between two characters that are the same size (e.g. the child and the woman geometrically scaled by a factor of 0.61 or a pair of one joint systems with

Quantity	Units	Geom. Scaling	Mass Scaling
Basic variables			
length	L	L	L
time	T	$L^{1/2}$	$L^{1/2}$
force	F	L^3	L^3
torque	FL	L^4	L^4
Motion variables			
displacement	L	L	L
velocity	LT^{-1}	$L^{1/2}$	$L^{1/2}$
acceleration	LT^{-2}	1	1
angular displacement	–	1	1
angular velocity	T^{-1}	$L^{-1/2}$	$L^{-1/2}$
angular acceleration	T^{-2}	L^{-1}	L^{-1}
Mechanical parameters			
mass	$FL^{-1}T^2$	L^3	L^3
stiffness	FL^{-1}	L^2	ML^{-1}
damping	$FL^{-1}T$	$L^{5/2}$	$ML^{-1/2}$
moment of inertia	FLT^2	L^5	L^5
torsional stiffness	FL	L^4	IL^{-1}
torsional damping	FLT	$L^{9/2}$	$IL^{-1/2}$

Figure 6: Scaling rules that capture differences in geometry, mass, and moment of inertia. The geometric scaling factor is derived assuming uniform scaling by factor L in all dimensions (geometric similarity), and assuming that the acceleration of gravity is invariant to scale. The mass scaling factor assumes in addition that mass scales by factor M and moment of inertia scales by factor I .

the same link lengths but different moments of inertia). Because the gains are scaled to keep angular terms the same for the two systems, the running stride of the two characters should have a similar appearance, despite the differences in moments of inertia. For example, if the leg lengths of two characters are equal, the leg extension during a stride and the stride frequency should be the same.

Equations 4 and 5 can be combined with the rules for geometric scaling to obtain a new set of rules that account for both size and relative moment of inertia differences. Using the geometric scaling rules in figure 6, we obtain the following equation to adjust gains for torsional stiffness and torsional damping:

$$k_1 = k_0 L^4 \left(\frac{I_1}{I_0 L^5} \right) = k_0 \left(\frac{I_1}{I_0} \right) L^{-1} \quad (6)$$

$$k_{v1} = k_{v0} L^{9/2} \left(\frac{I_1}{I_0 L^5} \right) = k_{v0} \left(\frac{I_1}{I_0} \right) L^{-1/2} \quad (7)$$

The combined set of rules for geometric scaling and scaling based on mass and moment of inertia properties



Figure 7: Image of a woman scaled to be the same height as a 3-year-old child.

is given in figure 6.

Applying equations 6 and 7 requires selecting the relevant body segments for each gain. The choice was clear in the case of the one-joint, one-body system but the human body model is much more complicated and the appropriate body segments are not trivially determined. Torque applied at a single joint drives the motion of several parts of the body, not just the two directly connected to that joint. Furthermore, the parts of the body affected by a particular joint change during different phases of the running cycle. For example, torque applied at the ankle affects primarily the motion of the foot during flight, but is responsible for propelling the entire body upward during stance. The flight gains for the ankle should scale with the length, mass, and moment of inertia of the foot while the stance gains for the ankle should scale with the length, mass, and moment of inertia of the entire body about the ankle.

For each gain in the control system, the appropriate moment of inertia was determined by identifying the parts of the body whose motion is significantly affected by that gain and summing their moments of inertia about the joint. For example, the gain responsible for swinging the hip forward during flight would be scaled by a factor dependent on the moment of inertia about the hip joint of the upper leg, lower leg, foot, and toe. The moment of inertia of each body part about the relevant joint $I_{B,J}$ is determined using the parallel axis theorem:

$$I_{B,J} = I_{B,com} + d_{B,J}^2 \quad (8)$$

where $I_{B,com}$ is the moment of inertia of the body about a line parallel to the joint axis and passing through the center of mass of the body, and $d_{B,J}$ is the distance from the center of mass of the body to the joint.

Although this scaling factor represents a far better

approximation than geometric scaling alone, it is still an approximation because the moment of inertia about a joint depends on the distance from the joint to the center of mass of each distal body part and changes over time as the joint angles change. For example, the moment of inertia of the leg about the hip is smaller when the leg is bent than when it is straight. However, this same approximation is included in the original control system for the runner because constant gains are used throughout flight to control the angle of the hip, although the knee bends and then straightens. To obtain a single scaling factor for each gain, we calculated the moments of inertia with the body standing upright in the position shown in figure 4.

The addition of scaling based on mass and moment of inertia can result in a substantial improvement over geometric scaling alone for characters with very different mass distributions, such as the woman and child. For example, a character generated by morphing the woman model one third of the distance toward the child model:

- does not run a single step when the control system of the woman is used without scaling or tuning,
- runs for 6 seconds with geometric scaling based on the ratio of leg lengths of the woman and the new model,
- runs with a stable gait (i.e. runs for at least 100 seconds) when geometric scaling and mass scaling are combined.

Tuning the Motion

Geometric and mass scaling provide a good approximation to a control system for a new human character, but these scaling methods are inexact and for characters that are significantly different the scaling will not provide a control system for steady-state running. To compensate for these approximations, we apply a final search step that fine-tunes the scaled control system to produce a desirable running motion.

To constrain the search process, we restricted the search to five high-level control parameters covering the most critical parts of the running motion: forward velocity, body attitude, and flight duration. One parameter, *ground speed matching*, controlled foot velocity relative to the ground at landing and affected the running speed. The second parameter controlled the desired pitch angle of the body during stance. Three parameters affected the duration of the flight phase: the timing of thrust and the extension of the ankle and knee during stance.

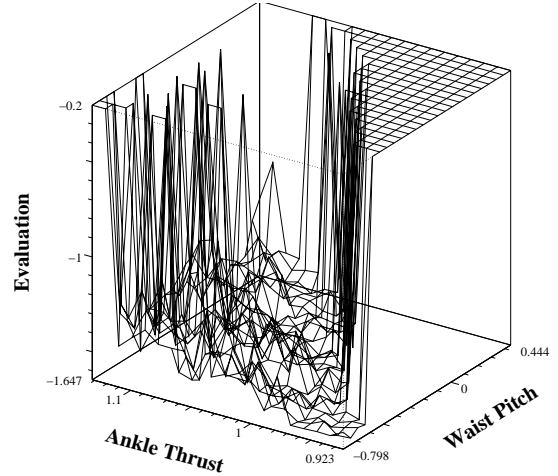


Figure 8: The evaluation function used to tune the runner control system has many local minima. This plot shows a sampling of the evaluation function as two of the tuning parameters responsible for body attitude and thrust are varied. The front corner of the figure shows a near optimal solution for this part of the search space. Values of -0.2 for the evaluation function indicate that the runner did not run for the desired running time.

The direction of the search was determined by an evaluation function designed to capture the desirability of the running motion. Values assigned to the search parameters defined a control system. This control system was used to control the runner for a run of a fixed duration (15 seconds in our experiments), and the resulting motion was evaluated for stability and balance, forward speed, body attitude, and desired flight duration. The evaluation function was also used to adjust the style of the running motion. For example, the evaluation function was adjusted to convert a stride where the feet of the runner were barely skimming over the ground to a stride that was more bouncy, to pull a runner out of a crouched running stride, and to eliminate a limp from a running character. Control of the desired value for body attitude also allowed us to produce a more appealing running motion.

The search space for this problem contains a large number of local minima. For example, figure 8 shows the evaluation function as two of the search param-

ters controlling body attitude and thrust are varied. The local minima in this plot result from the fact that the optimization function combines a variety of criteria that together provide a high-level evaluation of a complex motion. In an informal evaluation of search techniques, simulated annealing[22] appeared to provide better results on this search space than Powell's method, a line-search technique, and so simulated annealing was used to tune the control system for the characters described in this paper.

In both the case of the woman and child, the tuning process proceeded in stages. Scaling the control system of one character directly to reflect the parameters of a second character did not in general result in a stable running motion for the second character, which made the tuning process more difficult. To tune the control systems of the woman and child, we created sets of intermediate characters that allowed us to take smaller jumps between sets of physical parameters. The intermediate characters were then tuned in sequence. For example, in creating a desirable control system for the woman, control systems were first scaled and tuned for intermediate characters at 50%, 70%, and then 90% of the distance from the man to the woman. Similarly, two intermediate characters were used in tuning the control system of the child from that of the woman. We obtained good performance from the tuning process when a scaled, but untuned control system would allow a character to run for approximately 10 seconds without falling. Such a heuristic could be used to determine automatically which intermediate characters should be created. The step size could be decreased until the scaled character ran for a desired length of time without tuning. The control system for that intermediate character could then be tuned, and the tuning process could be continued from the intermediate character.

The results of scaling and tuning a control system for the man to control a woman and child are shown in figure 9, along with a sequence of video frames from a four-year-old child. All of the animated characters had a stable running motion, measured by verifying that they would run with a stable gait for at least 100 seconds.

Metamorphosis

The algorithm described here can be used not only for adapting control systems to achieve steady-state running but also to perform an on-line metamorphosis from one model to the other. To perform the metamorphosis, the graphical models, the physical models, and the control system are all interpolated to create a dynamic simulation that transitions from one model to another over a period of time. Figure 2 shows the transition

from a running man to a running woman over a period of 20 seconds.

The physical metamorphosis from the man to the woman was performed by creating nine, equally spaced dynamic models between the man and the woman and linearly interpolating the limb lengths, masses, and moments of inertia to create a new dynamic model for each simulation step. The initial control system was created in a similar fashion by interpolating between the scaled parameters for each of the nine intermediate models. This initial control system was not stable and the morphing runner fell over before it had completed the transition to the woman model. As in the adaptation of the steady-state control system, five parameters of the running motion were tuned to achieve stable running for the 20 second morph and steady-state running using the woman model for a subsequent 7 seconds. The initial values for the five parameters were those that were found by tuning the running motion of the man model. The tuning process then determines a rate of change for each parameter, creating a linear ramp for each parameter as the dynamic system transitions from the man to the woman.

The graphical metamorphosis was performed by shrinking a cube or cylinder onto the polygonal model for each body part of the man and the woman. The models that are created in this fashion have the same number of vertices and can be linearly interpolated to create a model for each frame of the animation. This simple algorithm for three-dimensional morphing is not as elegant as algorithms published in the literature (see, for example, [15]) but provides acceptable results because the polygonal models for the body parts are generally convex and because there is a close correspondence between the body parts of the man and the woman.

The metamorphosis shown in figure 2 was performed over a 20 second period. We would like to be able to morph between models more quickly, but faster physically realistic metamorphosis is difficult for several reasons. The primary difficulty is that the dynamic system, while physically realistic at each moment in time, is changing in a way that violates physical laws. For example, the body is changing in mass during the flight phase so angular momentum is not conserved. Similarly, while the foot is on the ground, the ground contact forces are applied to a constantly changing model. Physically realistic morphing of the control system is difficult because the running is not steady-state. The control system has step-to-step goals of maintaining forward speed, flight duration, and balance and immediate goals of moving the joints to the right angles. When the dynamic model is continuously

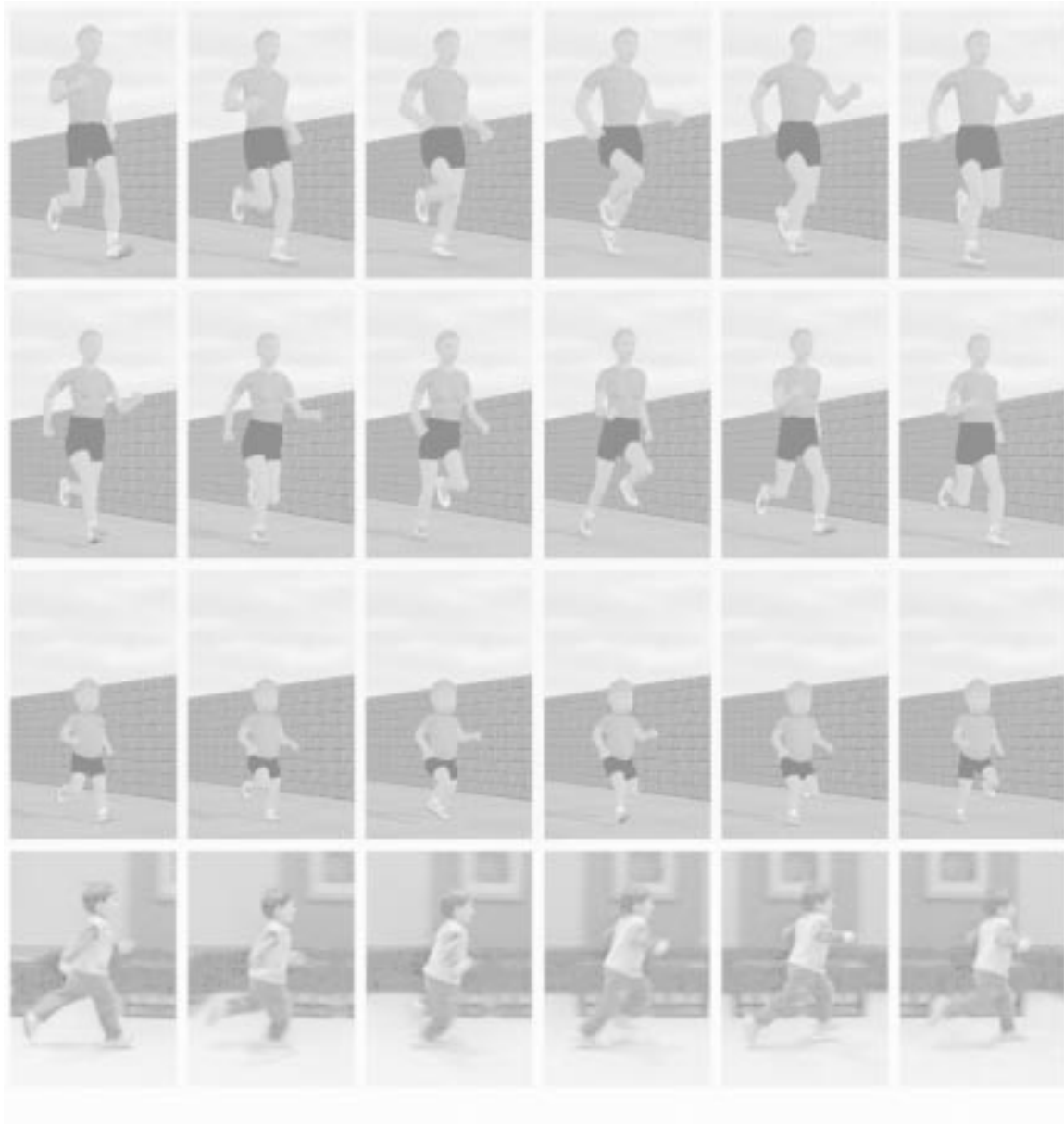


Figure 9: Images of a running man, woman, and child. The human child weighs 20 kilograms and is 1.07 meters tall. In each case, the spacing of the images is 0.066 seconds.

Quantity	Man	Woman	Child
Speed (m/s)			
human	2.58	2.39	2.13
simulated	4.41	4.45	2.52
Step Frequency ($steps/s$)			
human	2.83	2.45	3.8
simulated	3.18	3.33	4.3
Step Length (m)			
human	0.97	0.92	0.56
simulated	1.39	1.33	0.58
Vertical Lift (m)			
human	0.088	0.089	0.036
simulated	0.045	0.046	0.038

Figure 10: A comparison of data from the biomechanical literature with data recorded from the simulated runners. The velocity of the simulated man and woman is higher than in the reported experiments resulting in a higher step frequency, longer step length, and greater vertical lift. The match between the human child and the simulated child is within the variability between subjects.

changing, these two sets of goals and the choice of corresponding gains are no longer synchronized. For example, forward speed is controlled by the position of the foot at touchdown with the new desired speed being achieved by liftoff. When the dynamic model is morphing, the system in effect at the time the desired foot position angle was selected is no longer active when the new speed is achieved resulting in errors in the control of forward speed.

Discussion

One goal of this research is to demonstrate that simulations of humanlike motion can be automatically adapted to new dynamic models while maintaining the important properties of the running motion. Figure 9 shows a comparison of video footage of a human child and images of the simulated child. The human child has more variability in his motion than is seen in the motion of our simulated child and a slightly lower stride frequency. Figure 10 shows a comparison of biomechanical data from the literature[4, 11] with measurements of the simulated runners.

This paper presents algorithms that allow an animator to generate running motion for several different dynamic systems in an automatic fashion. By dividing the algorithm into two stages, scaling and tuning, we

chose to use a hybrid approach based on both explicit knowledge about the system and on automatic search. We made this decision because of our intuition that a fully automatic approach that attempted to tune all of the 88 control gains would not be successful. Similarly, we felt that a fully knowledge-based approach would fail because our understanding of human running, as represented by the control laws, is far from complete. The tuning process allows for some imprecision in the units and exact form of the control laws by adjusting several important parameters for an individual model.

To be widely applicable, this approach to adapting control systems must be independent of the particular behavior that we chose for our experiments. Would the same approach work for adapting control systems for diving, vaulting and bicycling to new models? Both the scaling laws and the selection of parameters for tuning are based on information about the control system and behavior. Some of that information is easy to determine from the control system. For example, geometric scaling requires that we know the units of each gain in the control system, but that information can easily be determined by examining the units of each control equation. Scaling based on mass and moment of inertia requires identifying the bodies that are most affected by a particular joint gain. We assumed that the lighter of the two bodies (or chains of bodies) would be most affected by the joint motion except when that body was in contact with the ground. Only the selection of parameters for tuning is specific to the running behavior. We chose higher-level parameters that we thought best represented the important properties of running: forward speed, flight duration, and balance. In principle, it should be possible to pick a similar set of parameters for other behaviors.

To be useful as part of a modeling and animation package, an algorithm for automatically adapting control systems must be robust for a wide variation in models. The child is markedly different from the woman or man but there are certainly parameter changes that are too extreme to be adjusted for by the combination of scaling laws and tuning presented here. For example, creatures whose physique demands a fundamentally different style of bipedal running could not be controlled using this approach. For example, birds and bipedal dinosaurs are “toe-strike” runners rather than “heel-strike” runners. The control laws for toe running probably differ not only in parameter values but also in structure from the control laws for heel-strike running.

Acknowledgments

The authors would like to thank Elizabeth de Goursac for her help in developing the early versions of the tools used in this paper and James O'Brien for the use of his rendering and modeling software. This project was supported in part by NSF NYI Grant No. IRI-9457621, by Mitsubishi Electric Research Laboratory, and by a Packard Fellowship.

References

- [1] J. Auslander, A. Fukunaga, H. Partovi, J. Christensen, L. Hsu, P. Reiss, A. Shuman, J. Marks, and J.T. Ngo. Further experience with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics*, 14(4):311–336, October 1995.
- [2] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, 1993. ISBN 0-19-507359-2.
- [3] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.
- [4] Y. Bhambhani and M. Singh. Metabolic and cinematographic analysis of walking and running in men and women. *Medicine and Science in Sports and Exercise*, 17(1):131–137, 1985.
- [5] Lynne Shapiro Brotman and Arun N. Netravali. Motion interpolation by optimal control. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 309–315, August 1988.
- [6] A. Bruderlin and T. W. Calvert. Goal-directed, dynamic animation of human walking. In *Computer Graphics (SIGGRAPH '98 Conference Proceedings)*, volume 23, pages 233–242, July 1989.
- [7] A. Bruderlin and L. Williams. Motion signal processing. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 97–104. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06–11 August 1995.
- [8] Armin Bruderlin and Tom Calvert. Interactive animation of personalized human locomotion. In *Proceedings of Graphics Interface '93*, pages 17–23, Toronto, Ontario, Canada, May 1993. Canadian Information Processing Society.
- [9] Michael F. Cohen. Interactive spacetime control for animation. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 293–302, July 1992.
- [10] W. T. Dempster and G. R. L. Gaughran. Properties of body segments based on size and weight. *American Journal of Anatomy*, 120:33–54, 1965.
- [11] V. L. Fortney. The kinematics and kinetics of the running pattern of two-, four-, and six-year-old children. *Research Quarterly for Exercise and Sport*, 54(2):126–135, 1983.
- [12] Michael Girard and Anthony A. Maciejewski. Computational modeling for the computer animation of legged figures. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 263–270, July 1985.
- [13] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *SIGGRAPH '95 Conference Proceedings*, Annual Conference Series, pages 71–78. ACM SIGGRAPH, Addison Wesley, August 1995.
- [14] P. S. Huang and M. van de Panne. A planning algorithm for dynamic motions. In *7th Eurographics Workshop on Animation and Simulation*, pages 169–182, 1996.
- [15] James R. Kent, Wayne E. Carlson, and Richard E. Parent. Shape transformation for polyhedral objects. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 47–54, July 1992.
- [16] J. Laszlo, M. van de Panne, and E. Fiume. Limit cycle control and its application to the animation of balancing and walking. In *SIGGRAPH '96 Conference Proceedings*, Annual Conference Series, pages 155–162. ACM SIGGRAPH, ACM Press, August 1996.
- [17] S. Lien and J. T. Kajiya. A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *IEEE Computer Graphics and Applications*, 4(10):35–41, 1984.
- [18] Z. Liu and M. Cohen. Decomposition of linked figure motion: Diving. In *5th Eurographics Workshop on Animation and Simulation*, 1994.
- [19] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 35–42. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

- [20] T. A. McMahon. *Muscles, Reflexes, and Locomotion*. Princeton University Press, Princeton, 1984.
- [21] J. Thomas Ngo and Joe Marks. Spacetime constraints revisited. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 343–350, August 1993.
- [22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, 1992.
- [23] Marc H. Raibert and Jessica K. Hodgins. Animation of dynamic legged locomotion. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 349–358, July 1991.
- [24] Robert Ringrose. Simulated creatures: Adapting control for variations in model or desired behavior. M.s. thesis, Massachusetts Institute of Technology, 1992.
- [25] C. F. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen. Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 155–162. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-9 August 1996.
- [26] D. E. Rosenthal and M. A. Sherman. High performance multibody simulations via symbolic equation manipulation and kane’s method. *Journal of Astronautical Sciences*, 34(3):223–239, 1986.
- [27] Karl Sims. Evolving 3d morphology and behavior by competition. In *Artificial Life IV*, pages 28–39, 1994.
- [28] Karl Sims. Evolving virtual creatures. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [29] Symbolic Dynamics Inc. *SD/Fast User’s Manual*. 1990.
- [30] M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 91–96. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [31] M. van de Panne, E. Fiume, and Z. Vranesic. Optimal controller synthesis using approximating-graph dynamic programming. In *Proceedings of the American Control Conference*, 1993.
- [32] M. van de Panne and A. Lamouret. Guided optimization for balanced locomotion. In *Eurographics Workshop on Computer Animation and Simulation '95*, pages 165–177, 1995.
- [33] Michiel van de Panne and Eugene Fiume. Sensor-actuator networks. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 335–342, August 1993.
- [34] Michiel van de Panne, Eugene Fiume, and Zvonko Vranesic. Reusable motion synthesis using state-space controllers. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 225–234, August 1990.
- [35] A. Witkin and Z. Popović. Motion warping. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 105–108. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [36] Andrew Witkin and Michael Kass. Spacetime constraints. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 159–168, August 1988.
- [37] X. Zhao, D. Tolani, B. Ting, and N. I. Badler. Simulating human movements using optimal control. In *Eurographics Workshop on Computer Animation and Simulation '96*, pages 109–120, 1996.